Folder structure of an Android App

# AndroidManifest.xml

This file contains information that devices needs to run the app. This xml file acts as an intermediate between android OS and our application.

When a user taps on apps icon, android OS will firstly reach this file then using the information available in the file system decides what to do next, what activity to open first.

# Myapplication folder

Contains all the source code file

# Myapplication folder (androidTest)

Contains test files and use it to create classes related to instrumented testing which are test involved with the device

# Myapplication folder (test)

Contains unit test cases

# Java folder

Contains classes all the time android studio will generate lot of classes for us to make it easier whenever we use supporting libraries like chrome retrofit to save our time.

# Res folder

Contains all non-code resources such as layout files, string resources, bitmap images.

* Drawable folder: contains all the different types of images we used for the development of the application every time we add a new image file.
* Layout folder: contains all the xml layout files
* Mipmap folder: contains images but different icon files used in them
* Value folder: contains xml file to hold different default values of the project like colors, strings
* Theme folder: contains details about the theme of the project

# Gradle Scripts

This will automatically build the system. We don’t have to download different library and put them to different folder.

Kotlin Language

1. Variables
   1. Var: Multiple values – mutable variable: muse have a type annotation or be initialized
   2. Val: Single Value – immutable variable
2. Function that is not return anything
   1. Formal way: fun showGreeting(): Unit{}
   2. Informal way: fun showThanks(){}
3. Function that will return value and body has only one line code
   1. Full way:

fun calculate(firstNumber: Int, secondNumber: Int): Int{

return firstNumber + secondNumber

}

* 1. Short way:

fun calculate2(num1: Int, num2: Int) = num1 + num2

1. Function with default parameter

fun showGreeting(name: String, city: String = "Regina"){

println("Welcome $name from $city!")

}

1. List
   1. Immutable List: cannot modify or add element
      1. List of String: val fruits: List<String> = listOf(“Banana”, “Apple”, “Orange”)
      2. Size of list: fruits.size
      3. Loop through the list: fruits.forEach{fruit -> println(fruit)}
   2. Mutable List
      1. List of String: val fruits: MutableList<String> = mutableListOf(“Banana”, “Apple”, “Orange”)
      2. Add element: fruits.add(“Mango”)

# Android Activity LifeCycle

[The activity lifecycle  |  Android Developers](https://developer.android.com/guide/components/activities/activity-lifecycle)

Diagram

Description automatically generated

When we run the application or we install and open this application in the phone. The android system will create an instance of this mainactivity.kt class in the memory that can be considered as the birth of the main activity instance.

For example:

* User takes a call -> The system has to move current activity of app to post state and move the app to the background.
* After the user finishes the phone call -> The system will take a web to foreground and transit the activity from pause state back to the resume state.

When we launch an activity:

* Firstly, it comes to create state. At that time, the android system will invoke activity instances on create callback function. In the oncreate function, we perform basic application startup logic that should happen only once for the entire life of the activity which is the must-have function. That why when we create a new activity class, android studio creates this function. This is the one create overridden function of the main activity. Overriding non-create function is compulsory, you must have a non-create function in an activity and this is the only compulsory lifecycle function. All other lifecycle functions are optional and we don’t have to override them unless we need to use them .
* When creating an activity, we extend AppCompatActivity class. It is extended from Android.app.Activity which has original life cycle callback functions. When the system creates an instance of one of our activity, in the memory system also creates instances of these parent classes. So if we haven’t overridden an optional lifecycle function like only start or on receive system will only knock activity class’s original function after the oncreate method finishes execution the activity enters to the started state and the system calls the onStart and onResume in a quick succession.
* The only start function of the activity class prepares the activity to enter the foreground and become interactive this is where the app initializes the code that creates the user interface. Then, during the resumed state app comes to the foreground, this is the state in which the activity interacts with the user. Now we can call activities running when that happen, onResumed function will be called activity stays in the resumed state until something happens to take focus away from it.
* When user navigates into another activity, activity will transit to pause state and one pause function will be invoked. If the activity returns to the resumed state, the system once again calls onResume method. So if our app required something to initialize, we should override the onResumed function. In professional projects, we may write codes inside onResumed overridden function for task like reconnecting with the server to load media files or initialize the camera. The system calls the onPaused function and take the activity to Pause state as soon as the user is leaving.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [java.lang.Object](https://developer.android.com/reference/java/lang/Object.html) | | | | | | | |
| ↳ | [android.content.Context](https://developer.android.com/reference/android/content/Context.html) | | | | | | |
|  | ↳ | [android.content.ContextWrapper](https://developer.android.com/reference/android/content/ContextWrapper.html) | | | | | |
|  |  | ↳ | [android.view.ContextThemeWrapper](https://developer.android.com/reference/android/view/ContextThemeWrapper.html) | | | | |
|  |  |  | ↳ | [android.app.Activity](https://developer.android.com/reference/android/app/Activity.html) | | | |
|  |  |  |  | ↳ | [androidx.activity.ComponentActivity](https://developer.android.com/reference/androidx/activity/ComponentActivity) | | |
|  |  |  |  |  | ↳ | [androidx.fragment.app.FragmentActivity](https://developer.android.com/reference/androidx/fragment/app/FragmentActivity) | |
|  |  |  |  |  |  | ↳ | [androidx.appcompat.app.AppCompatActivity](https://developer.android.com/reference/androidx/appcompat/app/AppCompatActivity) |

07-01-2023: Youtube – 2:50:20